

به نام خدا

استاد عبادزاده

علی ریشه 9531036

توضیح الگوریتم:

در فایل main شروع به خواندن داده ها می کنیم (لاین های اضافی حذف شده) حال باید ابتدا مرکز داده هارا بدست آوریم ،

Fuzzy c mean تمام ui ها (میزان تعلق) و vi (مرکز داده) را بدست می آوریم :

```
# Randomly initialize centers
centers = data[np.random.choice(
    data.shape[0], size=n_clusters, replace=False
), :]
# Compute initial distances
# Zeros are replaced by eps to avoid division issues
dist = np.fmax(
    cdist(centers, data, metric='sqeuclidean'),
    np.finfo(np.float64).eps
)
```

میتوانیم با تنظیم n_init و max_iter شانس رسیدن به مراکز با دقت موردنظر را بیشتر کنیم. حالت default به نتیجه میرسد.

حال ماتریس کواریانس را بدست می آوریم:

```
def calculate_covariance_mat(data, centers, U, m):
    C = []
    centers = np.array(centers)
    for i in range(len(centers)):
        shape = (2,2)
        divided_sum = np.zeros(shape)
        divisor_sum = .0
        for k in range(len(data)):
            divided_sum += np.power(U[i][k],m)* np.multiply(
                np.array([data[k] - centers[i]]), np.transpose(
                    np.array([data[k] - centers[i]])
                )
            )
        divisor_sum += np.power(U[i][k], m)
    C.append(divided_sum/ divisor_sum)
    return C
```

با بدست آمدن ماتریس کواریانس می توانیم G را بدست آوریم. در اینجا یک بار G را با داده های آموزشی بدست می آوریم و W را آموزش می دهیم.

در مرحله بعد G' را بر حسب داده های تست بدست می آوریم و نمودار داده هارا میکشیم.

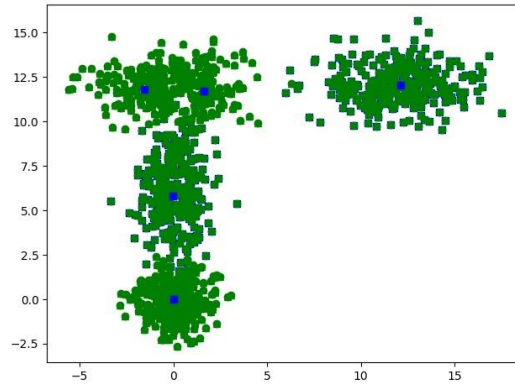
توضیح نمودار ها :

دو نمودار داریم ، اولی تمام داده ها و مراکز بدست آمده با fcm را نشان می دهد. مراکز: آبی - نقاط: سبز

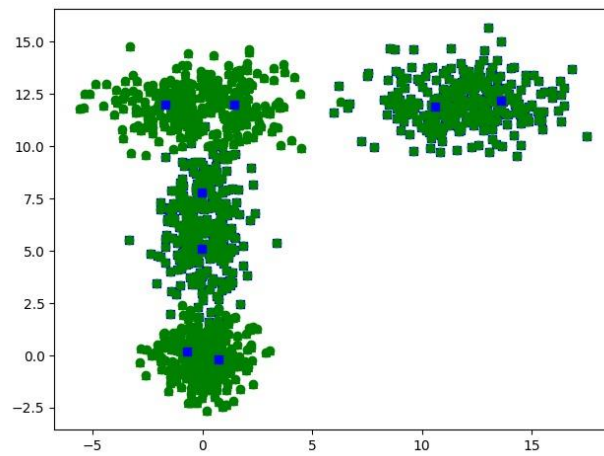
نمودار دومی کل نقاط ، نقاط درست حدس زده شده و غلط هارا نشان می دهد. درست: آبی - نقطه: سبز - غلط: قرمز

تست ها :

انتخاب 5 نقطه به عنوان مرکز :



مراکز با 8 نقطه :



نمودار با نقاط اشتباه و درست با 5 مرکز و شعاع 0.1 : در این تست accuracy داده های آموزشی 97 و تستی 84

